

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.021

«До захисту допущено»
Завідувач кафедри
О.В. Коваль
(ініціали, прізвище)
« » 2019р.

Магістерська дисертація

зі спеціальності - 121 Інженерія програмного забезпечення
за спеціалізацією - Інженерія програмного забезпечення розподілених систем
на тему: «Web-застосування для системи аналізу рейтингу викладацького складу
кафедри»

Виконав (-ла): студент (-ка) 6 курсу, групи ТВ-82мп
Кисіль Владислав Сергійович
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник к.т.н., доцент, Коваль О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019

**Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В.
(прізвище, ініціали) _____ (підпис)
« ____ » _____ 2019р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Кисілю Владиславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації “ Web-застосування для системи аналізу рейтингу викладацького складу кафедри ”

Науковий керівник доц., к.т.н., Коваль О.В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 4 ” листопада 2019 року № 3812-с

2. Строк подання студентом дисертації 9 грудня 2019 р. _____

3. Об'єкт дослідження рейтинг викладацького складу кафедри

4. Предмет дослідження системи аналізу рейтингу викладацького складу кафедри

5. Перелік питань, які потрібно розробити _____

1) проаналізувати особливості аналізу рейтингу викладацького складу кафедри;

2) проаналізувати стан систем аналізу рейтингу викладацького складу кафедри;

3) розробити алгоритмічну модель для розв'язання задачі аналізу рейтингу викладацького складу кафедри;

4) розробити веб-застосунок для системи аналізу рейтингу викладацького складу кафедри.

6. Орієнтований перелік ілюстративного матеріалу _____

1) Архітектурна модель

2) Етапи роботи з програмою

3) Функції програмного забезпечення

4) Структура програмного забезпечення

5) Інтерфейс

8. Дата видачі завдання « 28 » вересня 2019р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. – 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 – 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 – 29.03.19 р.	
6	Розробка програмного продукту	03.06 – 25.10.19 р.	
5	Переддипломна практика	02.09 – 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 – 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	18.12.19 р.	

Студент

(підпис)

Кисіль В.С.
(прізвище та ініціали)

Науковий керівник

(підпис)

Коваль О.В.
(прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи

Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 36 найменувань, одного додатку і містить 9 рисунків, 22 таблиці. Повний обсяг магістерської дисертації складає 74 сторінки.

Актуальність теми. Важко знайти людину яка б не користувалась сучасним додатками в інтернеті. Годі й говорити про спадаючу популярність настільних додатків та поширеність браузерів на всіх типах пристроїв із доступом до мережі інтернет.

Актуальність даної роботи підтверджується необхідністю у швидкому доступі до системи засобів перегляду рейтингу викладачів кафедри, яка реалізує в собі як можливості аналізу рейтингу викладачів за різні навчальні роки, так і можливість для внесення нових даних для доповнення бази даних, а також їх експортування у вигляді звітів у форматі документів для друку. Тому, враховуючи, що зараз робота по збору та аналізу даних виконується працівниками кафедри абсолютно в ручному режимі - розробка подібної системи є напрочуд актуальною і така система може мати великий попит в масштабах всієї навчальної установи.

Мета дослідження полягає в розробці високонавантаженої системи та підвищення ефективності процесів на кафедрі, пов'язаних із аналізом рейтингу викладачів, що дозволить зменшити час на їх генерацію та підвищити якість звітів, розширити функціональність вже існуючих стандартних систем.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати особливості збору рейтингових показників;
- проаналізувати стан використання інженерних та програмних рішень для систем аналізу рейтингу викладацького складу кафедри;
- розробити алгоритмічну модель для розв'язання задачі;
- розробити програмне забезпечення для перегляду рейтингу викладацького складу кафедри, генерації нових звітів по рейтингах.

Об'єктом дослідження є процес обробки даних для аналізу рейтингу викладацького складу кафедри.

Предметом дослідження методи та засоби розробки веб-системи для аналізу, перегляду та експорту рейтингів викладацького складу кафедри.

Методи дослідження. Розв'язання поставлених задач виконувались з використанням наступних методів:

- метод збору інформації для формування рейтингу за допомогою електронного збору через мережу Інтернет;
- метод рівномірного розподілення навантаження динамічного розрахунку рейтингу викладацького складу в реальному часі.

Найбільш суттєвими результатами магістерської дисертації є:

- запропоновано систему аналізу рейтингу викладачів кафедри, яка на відміну від існуючих є повністю автоматизованою та незалежна від платформи для запуску;
- система розроблена із архітектурним урахуванням можливого масштабування в майбутньому та розширення функціоналу;
- вперше серверна частина подібного продукту має API, що дозволяє інтегрувати систему до інших програмних продуктів;
- було запропоновано перенести статистику попередніх років до бази даних, таким чином розширюючи кількість даних для аналізу, що може знадобитись для генерації розширених звітів у майбутньому.

Практичне значення одержаних результатів роботи полягає в розробленому пакеті прикладних програм, направлених на ефективне та контрольоване використання потужностей використовуваних систем з метою реалізації висиконавантажуваної системи аналізу викладацького складу кафедри.

Ключові слова. ВЕБ, ЗАСТОСУНОК, WEB-ЗАСТОСУНОК, СИСТЕМА, АНАЛІЗ, РЕЙТИНГ, ВИКЛАДАЧ, КАФЕДРА.

ABSTRACT

Structure and volume of the dissertation

Master's dissertation consists of an introduction, five sections, a conclusion, a list of links of 36 titles, one application and contains 9 figures, 22 tables. The full volume of the master's thesis is 74 pages.

Significance of the topic. It is difficult to find a person who would not use modern applications on the Internet. The declining popularity of desktop applications and the prevalence of browsers on all types of devices with Internet access.

The relevance of this work is confirmed by the need for quick access to the faculty's rating review system, which implements both the ability to analyze the rating of teachers for different academic years, as well as the ability to enter new data to supplement the database, as well as export them in the form of reports in document format for printing. Therefore, given that the work of collecting and analyzing data is now done by the staff of the department completely in manual mode - the development of such a system is surprisingly relevant and such a system can be in great demand across the entire educational institution.

The **objective** is to develop a high-throughput system and increase the efficiency of the department's processes related to teacher rating analysis, which will reduce the time for their generation and improve the quality of reports and enhance the functionality of existing standard systems.

In order to achieve the objective next **goals** were defined:

- analyze the features of rating collection;
- to analyze the state of use of engineering and software solutions for systems of analysis of rating of teaching staff of the department;
- develop an algorithmic model for solving the problem;
- to develop software for reviewing the teaching staff rating of the department, generating new rating reports.

The **object** of the exploration is a process of data processing for analyzing the teaching staff of the department.

The **subject** of the exploration is methods and tools for developing a web-based system for analyzing, reviewing and exporting the teaching staff of the department.

Exploration methods. Solution of defined goals was made with usage of next methods:

- a method of collecting information for rating through electronic collection via the Internet;
- method of uniform load distribution of dynamic calculation of the teaching staff rating in real time.

The most valuable results of master's dissertation are:

- the system of analysis of rating of teachers of the department is offered, which unlike the existing ones is fully automated and independent of the launch platform;
- the system is designed with architectural consideration of possible future scaling and expansion of functionality;
- for the first time, the server side of such a product has an API that allows you to integrate the system with other software products;
- it has been proposed to migrate previous years' statistics to the database, thus expanding the amount of data for analysis that may be required to generate advanced reports in the future.

Practical value of the work that consists in the development of a package of applications aimed at the efficient and controlled use of the capacities of the systems used in order to implement a highly loaded system of analysis of the teaching staff of the department.

Key words. WEB, APPLICATION, WEB APPLICATION, SYSTEM, ANALYSIS, RATING, TEACHER, DEPARTMENT.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Особливості методики формування викладьких рейтингів.....	12
1.2 Особливості розробки серверної та веб-частини системи.....	17
2 АРХІТЕКТУРНА МОДЕЛЬ СИСТЕМИ АНАЛІЗУ РЕЙТИНГУ ВИКЛАДАЦЬКОГО СЛАДУ КАФЕДРИ.....	18
2.1 Загальний опис архітектури системи.....	18
2.2 Опис багатоваршавової архітектури системи.....	20
3 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ.....	22
3.1 Програмний засіб «АІС. Визначення рейтингу НПП»	22
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ.....	24
4.1 Особливості платформи ASP.NET Core та WebAPI	24
4.2 Фреймворк Angular та мова TypeScript	25
4.3 Сервер балансування NGINX.....	27
4.4 Особливості архітектури REST API	29
4.5 Опис моделі даних системи.....	32
4.6 Опис компонентів клієнтської та серверної складової	34
4.7 Графічний інтерфейс користувача.....	36
5. РОЗРОБКА СТАРТАП ПРОЕКТУ	41
5.1 Опис ідеї проекту.....	42
5.2 Технологічний аудит ідеї проекту	46
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	48
5.4 Аналіз ринкової стратегії проекту	56
5.5 Розроблення маркетингової програми стартап-проекту.....	60
ВИСНОВКИ	66
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

НПП	- науково-педагогічний працівник;
API	- програмний прикладний інтерфейс;
AIC	- автономна інформаційна система;
AJAX	- підхід до побудови інтерактивних веб-інтерфейсів;
REST	- representational state transfer;
SSL	- рівень захищених сокетів, криптографічний протокол;
IIS	- веб-сервер для Інтернет служб від Microsoft;
CRUD	- чотири базових функції для роботи із персистентним сховищем даних;
OC	- операційна система.

ВСТУП

В сучасному світі існує багато програмних додатків. Настільні програми встановлюються на персональному або робочому ком'ютері, в той час як веб-застосунки можна отримати через мережу Інтернет. Хоч обидва типи додатків базуються на основі програмного забезпечення, існують принципові відмінності між настільними програмами та веб-додатками.

Деякі користувачі стверджують, що веб-додатки є кращими по цілому ряду причин. Інші люди стверджують, що потужність настільних програм буде ще довго панувати на найвищому рівні. Але так чи інакше, всі люди використовують обидва типи програм у повсякденному житті.

Веб-застосунки є гнучким рішенням, що дозволяє отримати доступ до сервісу чи послуги всюди де є веб-браузер та підключення до мережі Інтернет. Також такий тип програм не потребує встановлення на користувацьке обладнання, та не потребує ручних оновлень, користувач може бути певен – у нього завжди найактуальніша версія програмного забезпечення. Саме тому було вирішено розпочати розробку програми, що буде працювати у вікні браузера, і буде не тільки повторювати вже існуючий функціонал аналогічних настільних систем, а й запропонує великий спектр можливостей для розширення функціоналу.

Метою дослідження магістерської дисертації є розробка веб-застосунку що дозволяє проводити моніторинг активності, формування на його підставі аналітичних рейтингів науково-педагогічних та наукових працівників (далі - Викладачів), докторів, аспірантів. Як результат, очікується, що це має мотивувати викладачів до наукової діяльності та підвищення результативності викладачів, а також а також підвищення конкурентоспроможності університету, забезпечення його присутності та сприяння покращенню позицій у міжнародних та відчизняних рейтингах. Магістерська дисертація містить у собі п'ять розділів. Перший розділ несе в собі опис задач розробки програмного продукту, для формування та аналізу рейтингу викладацького складу кафедри.

Другий розділ містить докладний опис моделі запропонованого рішення задачі аналізу рейтингу кафедри.

Третій розділ розглядає існуючий аналог розроблюваної програми, який існує у відкритому доступі, а саме його недоліки і переваги у порівнянні зі створеною системою.

У четвертому розділі йде мова про особливості реалізації веб-застосунку та докладна методика користування пороботі із системою, ілюстрації із графічним інтерфейсом програмного продукту.

П'ятий розділ розглядає стартап проект та методи його просування на конкурентний ринок.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Цивілізаційні зміни, що зараз переживає сучасний світ, кидають виклик та ставлять багато задач перед системою освіти. Реальність сьогодення визначає інноваційні підходи, що вимагають ретроспективно орієнтованої моделі оцінки завершених знань. Перед викладачами вищих навчальних закладів ставляться цілі їх позиціонування як джерелу продукування нових знань, технік, підходів та технологій.

Вирішення завдань освіти у нашій державі потребує вискокваліфікованих фахівців, що вміють якісно передавати знання та уміння студентам у процесі навчання. Адже підготовка нових висококваліфікованих фахівців напряму залежить від якості викладання дисциплін у вищому навчальному закладі, яка визначається в процесі аналізу та оцінювання ефективності педагогічних компетенції викладачів та педагогів.

1.1 Особливості методики формування викладських рейтингів

Діагностування педагогічної компетентності наставників і рейтингування викладачів вищих навчальних закладів є одним із пріоритетних завдань підвищення якості навчального процесу та отриманих професіоналів-працівників у майбутньому.

Світові рейтингові системи свідчать, що усі країни цікавляться рівнем результативності системи освіти. Вимірювати її можна по-різному, наприклад, за досягненнями студентів-випускників, або за тим, як використовується на ринку робоча сила. Інші аспекти результативності стосуються не тільки знань, а й суджень, розумінь, принципів та цінностей. Результативність характеризується не тільки кількісно, а і якісно, тому всі отримані статистики треба розглядати в ракурсі всіх аспектів, завдяки який існує та функціонує система освіти. Країни прагнуть

порівняти успішність у навчанні через звітність, бо приділяють цьому питанню чимало уваги, особливо до якості та стандартів.

Із відкритих джерел відомо, рейтингом називають числові, порядкові показники, які є мірами популярності або успішності, що відображають ефект та вплив об'єкта або явища. Також це може бути показником діяльності, популярності якоїсь особи, групи, програм в момент часу, визначається голосуванням, опитуванням чи іншими методами збору інформації, а визначається місцем, яке посідає.

У тлумачному словнику С. Ожегова під рейтингом розуміємо показник популярності особи і ступінь її популярності [3, 674].

Як зазначає Ю. Похолков, системним чинником будь-якого рейтингу є його мета, що визначає методологію рейтингу, структуру критеріїв і показників, процедуру збирання інформації та обробки даних. При цьому один і той самий викладач у рейтингу, що мають різну цільову спрямованість, може посідати абсолютно різні позиції [5, 10].

Ідея ранжування викладачів є досить давньою і практикується у країнах Західної Європи та світу. У Сполучених Штатах Америки до процесу оцінювання компетентності викладачів залучаються працівники освіти, експерти, адміністрація вищого навчального закладу, студенти та інші зацікавлені особи. Американські дослідники оцінюють педагогічну компетентність викладачів вищої школи за такими критеріями: планування навчального процесу; логічна послідовність вивчення навчального матеріалу; викладання навчальної дисципліни на професійному рівні; зрозумілість і доступність навчального матеріалу; науковий рівень педагога; використання новітніх технологій і сучасного навчального матеріалу; прозорість викладення думки викладача; об'єктивність у процесі оцінювання рівня знань студентів; вболівання викладача за ґрунтовні знання студентів; створення доброзичливої атмосфери в процесі навчання; зацікавленість студентів у вивченні наступного курсу з навчальної дисципліни через: систематичне підвищення педагогічної майстерності, прагнення до самовдосконалення, участь у

наукових конференціях, проведення громадських заходів, ставлення до роботи, заслуги в минулому, результативність використання окремих методів навчання тощо [2, 115–117].

Великі можливості, пов'язані із оцінкою викладацької роботи дає рейтингове оцінювання. Визначимо функції рейтингу викладачів вищих навчальних закладів. Його загальна складова, відкриті результати та супутня інформація про навчальний заклад дозволяє майбутнім студентам отримати більш чітку інформацію для аналізу. Рейтинг допомагає формувати імідж викладача. Він має стимулювати різні педагогічні навички: теорія, властивості, історія, інноваційні технології. Саме така мета є для впровадження рейтингової оцінки діяльності на кафедрі, факультеті, в інституті, коли запроваджуються подібні рішення. Всі викладачі будуть прагнути покращити свої показники, які на нього впливають.

Чітко проаналізувавши можливі переваги, можна зробити перший висновок: розумно складений рейтинг сприяє професійному й етичному зростанню всього викладацького складу універсистему. Але створення рейтингу є багатограним та складним процесом. Потрібно сформулювати критерії оцінювання та алгоритми із його формування. Чітко проаналізувавши процес оцінювання викладачів на кафедрі показав, що визначення рейтингу проходить у два етапи: 1-й етап – визначення ключових чинників оцінки роботи та встановлення її значущості; 2-й етап – визначення рейтингу професорсько-викладацького складу відповідно до системи оцінок його роботи. Варто зазначити, що сухий аналіз статистичних даних про працівників, облік навчально-методичної роботи не дає змоги виконати якісний аналіз кадрового складу кафедри.

Інформаційні системи, що на даний момент викоростовують в управлінні складом кафедр, дають можливість забезпечити об'єктивний підхід для визначення рейтингу викладачів та формуванню і підбору складу кафедри за різними якістьми. На рейтинг викладача та його діяльності впливають кількість виконаних навчальних годин; при оцінюванні методичної і наукової роботи домінуючу роль відіграє кількість опублікованих методичних рекомендацій та їх обсяг (у друкованих

аркушах), кількість статей, монографій, доповідей на наукових конференціях; результативність виховної роботи оцінюється за кількістю проведених виховних заходів (конкурсів, турнірів, екскурсій, бесід зі студентами тощо). Однак, лекційна майстерність викладача, його вміння застосовувати передові дидактичні технології, педагогічну техніку, власні методики навчання не беруться до уваги. У багатьох вищих навчальних закладах України при проведенні рейтингу викладачів ураховується їх інноваційна та міжнародна діяльність, презентація власного інноваційного досвіду викладання, проведення тренінгу для співробітників, участь у підготовці та реалізації міжнародного проекту, розробка комп'ютерного тесту, участь в організації заходів інноваційного спрямування чи підготовці наповнення сторінки порталу структурного підрозділу університету, впровадження викладачем науковотехнологічного процесу у виробництво, внесок викладача до матеріально-Педагогічні науки: теорія, історія, інноваційні технології, технічної бази: участь у створенні й переобладнанні лабораторій та аудиторій, відповідальність за їх стан, виконавчу і трудову дисципліну викладача, а також передбачають зменшення рейтингу викладача при порушенні ним трудової дисципліни, несвоєчасному виконанні розпоряджень, наказів, систематичному запізненні на заняття. [4]

Робочий час викладача складається із виконання ним трудових обов'язків: начальна, методична та організаційна робота (в тому числі і робота із забезпечення контролю якості підготовки фахівців).

Наказом ректора встановлюються та затверджуються норми часу на навчальну роботу, в Університеті використовують для планування й оцінки обсягу роботи, також враховуються норми на методичні, наукові й організаційні роботи, схвалені вченою радою вищого навчального закладу.

Норми на роботи, які є не зазначеними, визначає завідувач кафедри.

Сумарне навантаження викладача R обчислюється за формулою (1.1) :

$$R_{\text{нп}} = R_{\text{н-м}} + R_{\text{н-і}} + R_{\text{о-в}} \quad (1.1)$$

де $R_{н-м}$, $R_{н-і}$, $R_{о-в}$ – навантаження науково-педагогічного працівника (далі - НПП), в баллах, за виконання навчально-методичних, наукових та організаційних робіт відповідно.

НПП напередодні навчального року, з урахуванням зауважень та рекомендацій завідуючого кафедри, планує виконання кожного з видів робіт на навчальний рік так фіксує планові обсяги робіт в своєму індивідуальному плані. Кількість балів у сумі за виконання всіх запланованих робіт має становити не менше 1500 балів за навчальний рік. Обсяг навчального навантаження на викладача у годинах відповідає такій же кількості балів.

В кінці кожного навчального року викладач має фіксувати та вносити в свій індивідуальний план фактичні результати, обсяги виконаних робіт та проводить обчислення результатів своєї роботи.

Наприкінці навчального року викладач фіксує у своєму індивідуальному плані обсяги виконаних робіт по факту та обчислює результативність своєї роботи.

Рейтинг всієї кафедри обчислюється за наступною формулою (1.2) :

$$R_{kaf} = \frac{1}{N} \sum_i^n R \quad (1.2)$$

де N та n – кількість ставок НПП кафедри та кількість фізичних осіб НПП кафедри відповідно.

Рейтинг завідуючого кафедри можна розрахувати за формулою (1.3) :

$$R_{zk} = R + 0,3R_{kaf} \quad (1.3)$$

де R та R_{kaf} – рейтинг викладацької діяльності та рейтинг кафедри відповідно.

1.2 Особливості розробки серверної та веб-частини системи

Всім відомо, що архітектурні рішення являються фундаментом та закладами для будь-яких додатків. В тому числі для з високим навантаженням. Архітектура серверної частини веб-застосування закладає 90% успішності його роботи.

Важливо розібратись в принципах побудови більш детально. Складно передбачити, яке майбутнє чекає на застосунок. Будь-який аспект користування додатком може змінитись, наприклад, збільшення користувачів у системі або збільшення об'єму даних. Також немало важливим є фактор масштабованості та розширення функціоналу додатку. Можливі випадки, коли різко почне набирати популярність якийсь другорядний функціонал, що може спричинити нову ланку проблем. Зрозуміло, що чим більше стає система, тим більш складно буде передбачити або спланувати її оновлення у майбутньому. Успішність роботи в такому випадку матиме акценти не на конкретному уточненні майбутнього функціоналу, а на забезпечення гнучкості. Адже гнучкість дозволить вносити швидкі зміни та доповнення. Це найбільша важливість будь-якої системи.

Не потрібно намагатись передбачати обсяг аудиторії на роки вперед. Те ж саме стосується і архітектури. Стосується як і апаратної частини, так і програмної. Рішення мають бути поступовими і точними. Це буде гарантувати уміреність витрат і спрощення управлінням інфраструктурою. Краще всього, на етапі запуску проекту, використовувати хмари для хостингу нових проектів. Потенційно це дозволить знизити фінансову навантажку на проект та дозволить робити розширення фізичних ресурсів без шкоди для проекту.

Приватна мережа – послуга, що зараз користується популярністю, бо вона дозволяє використовувати апаратні ресурси серверів прямо в хмарі. Таким чином, є нагода скористатись можливістю масштабування додатку без переїзду на інше середовище. Краще робити рішення простими, це розробляти вкрай складно, але всі зусилля будуть окуповуватись все швидше і швидше при подальшій модернізації бекенду та веб-застосунку.

2 АРХІТЕКТУРНА МОДЕЛЬ СИСТЕМИ АНАЛІЗУ РЕЙТИНГУ ВИКЛАДАЦЬКОГО СЛАДУ КАФЕДРИ

Вибір архітектурних рішень задач є чи не найважливішим етапом під час розробки. Чато такі архітектурні рішення називають архітектурними стилями і парадигмами. До них відносяться такі шаблони як клієнт-сервер, компонентна архітектура, сервісо-орієнтована архітектура. Якість розроблених компонентів та рішень можуть стати вирішальним фактором успішності застосунку. Швидкодія систем сьогодні – важливий фактор під час обрання програм, що вирішують ту чи інші прикладну задачу.

2.1 Загальний опис архітектури системи

Багаторівнева клієнт-серверна архітектура описує розподілені системи, які складаються із окремого додатку для клієнта і сервера в об'єднуючій їх мережі. Найпростішою формою системи клієнт-сервер є двох-рівнева архітектура – це серверне програмне забезпечення, до якого звертаються безліч клієнтських додатків.

Історично така архітектура представляє собою додаток з графічним інтерфейсом, що обмінюється даними з файловим сервером або із сервером бази даних, на якому у формі процедур бази даних знаходиться частина робочої логіки. Якщо розглядати більш абстрактно, цей архітектурний стиль описує відношення між клієнтським додатком та сервером, де клієнт ініціює один або більше запитів, очікує відповіді та оброблює її при отриманні. Для зв'язку із клієнтом сервер може використовувати широкий діапазон протоколів і форматів даних.

На сьогоднішній день найкращим прикладом таких додатків служать веб-застосунки, що виконуються в Інтернеті або внутрішніх мережах організацій.

До різновидів стилю клієнт-сервер відносяться:

— Системи клієнт-черга-клієнт. В цьому підході клієнт може обмінюватись даними з іншими клієнтами через чергу на сервері. Клієнти можуть

читати та відправляти дані на сервер, який виступає у ролі простої черги для збереження даних. Завдяки такому підходу клієнти можуть розподіляти і синхронізувати файли та їх властивості;

- Однорангові додатки (Peer-to-Peer, P2P). Створені на базі клієнт-черга-клієнт, стиль P2P дозволяє клієнту та серверу обмінюватись ролями з ціллю розподілення та синхронізації файлів і даних між великою кількістю клієнтів. Ця схема дозволяє значно розширити моделю клієнт-сервер, додаючи кількісні відповіді на запити, сумісно використовувані дані, та знаходження ресурсів а також стійкість при зменшенні кількості учасників у мережі;

- Сервери додатків. Спеціальний архітектурний стиль, при якому додатки і сервіси розміщуються і виконуються на сервері, тонкий клієнт виконує операції доступу до них через браузер або спеціальне програмне забезпечення встановлене на стороні користувача. В приклад можна привести клієнт, що працює із сервером, через таке середовище як SignalR;

- Багатошарова архітектура. Архітектура в якій описано більш ніж два рівні – називається багатошаровою, в такій системі є інтерфейс користувача, шар бізнес-логіки, та шар доступу до даних. Використовуючи цю архітектуру, користувачі мають змогу робити різні запити через рівень користувацького інтерфейсу, який взаємодіє тільки із шаром бізнес-логіки. Шар користувацького інтерфейсу ні в якому разі не має мати доступу до даних, а також не повинен ніяк взаємодіяти із логікою ніяким іншим чином чи засобом. Таким чином кожен шар архітектури має свою відому відповідальність.

Основні переваги архітектурного стилю клієнт-сервер:

- Безпека даних. Всі данні зберігаються на сервері, який зазвичай може забезпечити більшу надійність та контроль безпеки, ніж звичайна настільна програма;

- Централізований доступ до даних. Оскільки дані зберігаються тільки на сервері, адміністрування такого ресурсу значно простіше, ніж в інших архітектурних стилях;

— Простота в обслуговуванні. Ролі і відповідальність обчислювальної системи розподілені між декількома серверами, що спілкуються один із одним через мережу. Завдяки цьому клієнтська частина гарантовано залишається відокремленою від внутрішніх подій, що відбуваються із сервером (ремонт, оновлення або переміщення).

2.2 Опис багатоварової архітектури системи

Було вирішено спроектувати додаток у стилі багатоварової архітектури. Вона підходить під усі критерії, для створення веб-застосунку та є дуже поширеною на даний момент.

Найбільш традиційна архітектура приведена на Рисунку 2.1

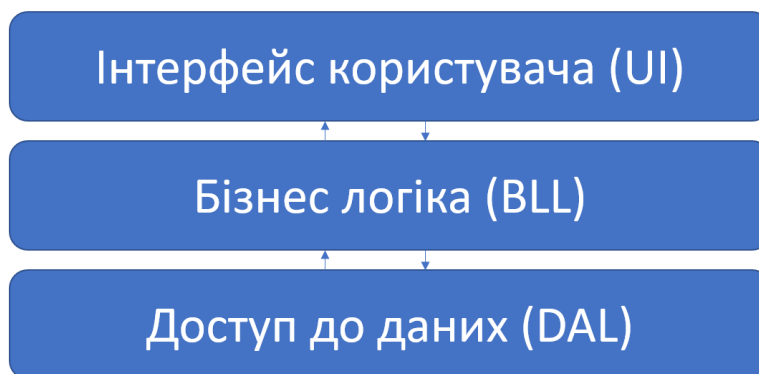


Рисунок 2.1 — Типова схема багатоварової архітектури.

Архітектурні шари часто скорочуються такими аббревіатурами як UI (інтерфейс користувача), BLL (шар бізнес-логіки) та DAL (рівень доступу до даних). Використовуючи цю архітектуру, користувачі робитимуть запити через користувацький інтерфейс, який взаємодіє лише з шаром бізнес-логіки. Бізнес-логіка, в свою чергу, може викликати DAL для запитів доступу до даних. Головна особливість полягає у тому, що шар користувацького інтерфейсу не має доступу до

рівня даних, а також не взаємодіє ніяким іншим чином із бізнес-логікою. Таким чином, кожен шар має свою певну відповідальність.

Одним із небагатьох недоліків цього традиційного способу відшарування є те, що в результаті, на етапі компілювання, програма буде збиратись зверху до низу. Тобто, шар бізнес логіки буде залежати від шару доступу до даних, тобто при тестуванні доведеться звертати увагу на деталі реалізації доступу до даних. Тестування бізнес-логіки в такій архітектурі часто буває складною, вимагає тестової бази даних. Саме такому випадку краще застосувати принцип інверсії залежності.

Модель архітектури серверної частини запропонована на рисунку 2.2.

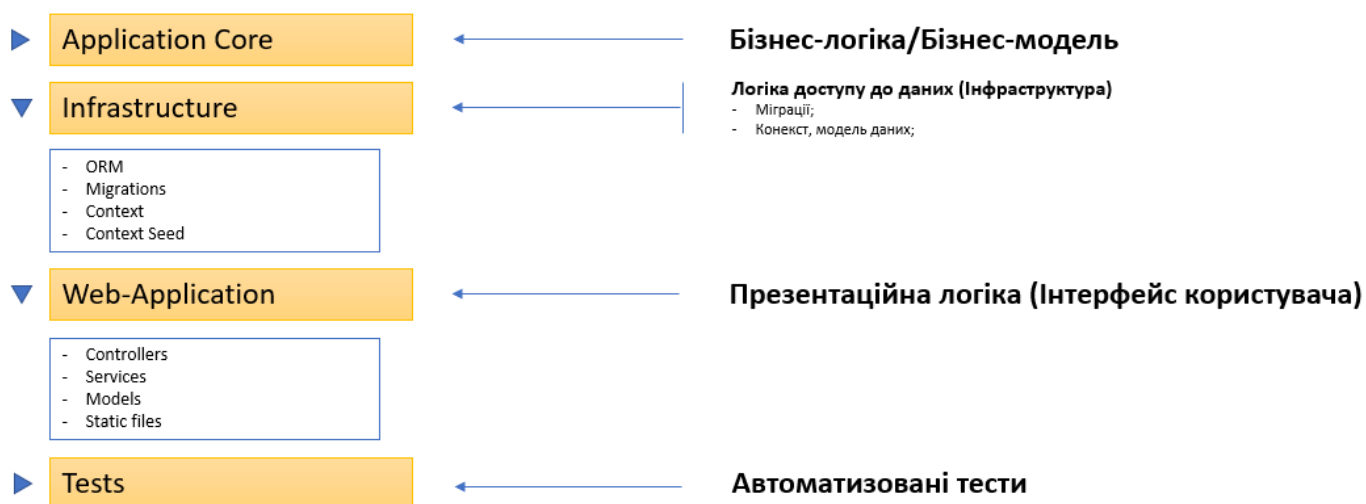


Рисунок 2.2 — Запропонована монолітна архітектура веб-застосунку з трьома проектами.

3 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ

На даний час в університеті вже існує ресурс, який дозволяє проводити аналіз рейтингу науково-педагогічних працівників. В цьому розділі будуть розглянуті його недоліки та переваги.

3.1 Програмний засіб «AIC. Визначення рейтингу НПП»

Згідно до опису та інформації на сайті виробника, система представляє собою програмний комплекс, що призначено для аналізу, обробки та оцівання роботи викладачів та інших наукових працівників, що дає змогу визначати, як діяльність тієї чи іншої людини впливає на вищий навчальний заклад в цілому. Дана система дозволяє вчасно виявляти недоліки у викладацькому складі і на підставі зібраної інформації впливатиме на покращення майбутніх результатів роботи вищого навчального закладу.

Стимулом для запровадження подібних рейтингових систем для науково-педагогічних працівників є:

- покращення показників результативності професійної діяльності НПП;
- об'єктивне оцінювання всього педагогічного складу та його діяльності, в усіх підрозділах вищого навчального закладу;
- гарантування здорової та правильної конкуренції, мотивація співробітників;
- підвищення кваліфікаційного рівня та рейтингу університету.

Рейтинг розрахунку в даній системі визначається наступним чином:

- розраховується абсолютний власний рейтинг НПП;
- розраховується сукупність рейтингів НПП, що є загальним рейтингом кафедри.

Виходячи із положення про порядок визначення рейтингу науково-педагогічних педагогічних працівників [1], отримана формула для розрахунку

рейтингу професійного складу науково-педагогічних працівників вищого навчального закладу.

Робота викладацького складу, на протязі всього учбового навчального року, допомагає визначити та провести оцінювальну діяльність на основі їх вкладу до підвищення та підтримання рейтингу та авторитету університету. Для підведення підсумків потрібно врахувати всі види діяльності НПП.

Система АІС "Визначення рейтингу науково-педагогічних працівників" була впроваджена в роботу 21.05.2009 р.

Роблячи висновок після ознайомлення із вже існуючим програмним продуктом і його порівняння із розроблюваним веб-застосунком, можна зробити висновок, що новий додаток матиме ряд переваг, він є незалежним від платформи для запуску, не потребує встановлення на комп'ютер, доступний цілодобово, є сумісний із файлами до старої системи, і, що найголовніше, має великий потенціал до розширення та модифікації.

В даному розділі було переглянуто переваги та недоліки настільного застосунку АІС "Визначення рейтингу науково-педагогічних працівників" і після порівняння можемо зробити висновок, що розроблюваний веб-застосунок має ряд переваг над вже існуючим.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

Для розробки системи було вирішено використовувати мови програмування TypeScript для клієнтської частини та .Net Core, ASP.NET Core WebAPI для серверної частини разом із сервером NGINX для забезпечення балансування при умовах високої завантаженості. Клієнтська частина, користувацький інтерфейс було розроблено із допомогою фреймворку Angular та SCSS. Взаємодія між веб-застосунком та сервером відбувається через API, із врахуванням кращих практик REST API.

Користувацький інтерфейс був розроблений за допомогою фреймворку Angular. Платформа є лідером серед своїх конкурентів, Angular добре документований, та є найпопулярнішим фронтенд фреймворком на даний момент.

4.1 Особливості платформи ASP.NET Core та WebAPI

У 2015 році Microsoft заявила про новий напрямок для ASP.NET і MVC Framework, що в підсумку призвело до появи інфраструктури ASP.NET Core MVC.

Платформа ASP.NET Core побудована на основі .NET Core, яка представляє собою міжплатформову версію .NET Framework без інтерфейсів програмування додатків, специфічних для Windows. Головною операційною системою як і раніше є Windows, але веб-додатки все частіше розміщуються в невеликих і простих контейнерах на хмарних платформах. За рахунок прийняття міжплатформного підходу компанія Microsoft розширила область охоплення .NET, зробивши можливим розгортання додатків ASP.NET Core на більш широкому наборі середовищ розміщення, а в якості бонусу надала розробникам можливість створювати веб-додатки ASP.NET Core на машинах Linux і macOS.

ASP.NET Core - абсолютно нова інфраструктура. Вона простіше, з нею легше працювати, і вона вільна від спадщини, що була присутня в технології WebForms. Базована на .NET Core, вона підтримує розробку веб-додатків для ряду платформ і контейнерів.

Інфраструктура ASP.NET Core MVC пропонує функціональність початкової інфраструктури ASP.NET MVC Framework, побудованої поверх нової платформи ASP.NET Core. Вона інтегрує функціональні засоби, такі як Web API, та підтримує більш природний спосіб генерування складного вмісту і робить основні завдання розробки, такі як модульне тестування простішим і передбачуванішим.

ASP.NET Core, у порівнянні зі старим .NET Framework має наступні переваги:

- ASP.NET Core має великий набір архітектурних змін, що в результаті роблять платформу більш зрозумілою до вивчення та модульною;
- ASP.NET Core більше не вимагає компоненту System.Web.DLL. Замість цього компоненту тепер є декілька дрібних пакетів із NuGet;
- вище вказані плюси дозволяють зменшити вагу додатку, завдяки включенню тільки тих пакетів, які потребує розробник;
- покращена безпека, швидкодія;
- ASP.NET Core додатки тепер можна запускати на Windows, Mac та Linux;
- .NET Core тепер підтримує версіонізацію;
- покращене середовище Visual Studio;
- сумісність із усіма хмарними хостингами;
- сумісність із IIS 7+ та IIS Express.

4.2 Фреймворк Angular та мова TypeScript

Фреймворк Angular 8 (старі назви: Angular.js, Angular 2+) — відкрита платформа для створення інтерфейсів користувача для всіх пристроїв, що мають

браузер із підтримкою JavaScript. Фреймворк Angular написаний на TypeScript та має реалізацію всіх базових компонентів, що потрібні для побудови повноцінного веб-застосунку, які можна імпортувати в будь-який зручний момент.

Додатки створені на основі Angular є модульними, ця модульна система називається NgModules.

Система NgModule це деревовидний набір компонентів. Кожен компонент має свій батьківський компонент. Головним компонентом в будь-якому Angular-додатку є кореневий модуль, який зазвичай називають AppModule, що дає змогу механізм запуску для програми.

Важливою особливістю даної платформи є те, що в Angular-додаток можна імпортувати код не тільки написаний мовою TypeScript, а й раніше написаний код JavaScript.

Кросплатформний Angular не намагається надати повний інструментар для створення повноцінного веб-застосунку. Він є вузьконаправленим, для створення користувацького інтерфейсу, і тому не має включати в себе безліч інструментів. Розробники самі мають вирішити, що має містити їх програма, та додати це до неї у вигляді модулю. Така гнучкість дозволяє обирати будь-які бібліотеки, які розробник обере по різному ряду причин.

Платформа має досить великий потенціал для створення великих та масштабованих веб-застосунків, що опрацьовують дані і змінюються із часом, без перезавантаження сторінок. Angular у своїй суті спонукає розробників робити програми швидкими, простими, відмово-стійкими. Він обробляє тільки користувацький інтерфейс у застосунках.

Подібна будова веб-застосунків є підвидом шаблону модель-представлення-контролер, і може бути імплементована у поєднанні із іншими JavaScript бібліотеками.

Важливою особливістю веб-застосунків є використання віртуальної моделі об'єктів документів або віртуального DOM. Angular обчислює відмінності між станом даних впродовж використання додатку, та заносить все це до кеш-пам'яті, а

коли вони з'являться — оновлює відображення дерева DOM у браузері. Ця властивість дозволяє розробляти додатки з ефектом, ніби вся весь інтерфейс реагує на кожну зміну, а Angular візуально оновлює тільки ті частини програми, стан компонентів яких було дійсно змінено.

Допоміжний додаток SASS, або SCSS — це препроцесор, синтаксична надстройка над CSS, що використовується разом із Angular, та дозволяє динамічно створювати та генерувати стилі CSS. Код написаний на SCSS на етапі компіляції буде інтерпретовано до звичайного CSS. SASS допомагає зменшити кількість коду, що повторюється і заощадити час витрачений на розробку для програмістів, це стабільний та потужний додаток для веб-розробників, що дозволяє створювати та документувати код просто, швидко, ефективно і зрозуміло. Серед недоліків використання даної надстройки є те, що для освоєння нової мови розробник має витратити час та взаємодія у великих командах може бути ускладнена.

4.3 Сервер балансування NGINX

Використання серверів балансування є досить важливим, адже через нього проходять усі запити користувачів. І саме сервер Nginx займається переадресацією web-запитів до основного серверу до серверів обробки запиту у реальному часі. Nginx є відкритим для безкоштовного використання, а також являється проксі-сервером, доступним для завантаження на усіх настільних платформах. Є версії для Microsoft Windows, macOS та Linux x64.

Даний продукт є в розробці з 2002-го року, на замовлення Rambler і постійно розвивається. У 2004 році було опубліковано пілотний реліз для ознайомлення. В Nginx робочі процеси обслуговують одночасно безліч веб-з'єднань, опрацьовуючи їх викликами операційної системи, такими як select, epoll і kqueue, в системах Microsoft Windows, Linux та FreeBSD відповідно. Обробкою дескрипторів займаються процеси пов'язані з циклами подій. Дані, що отримані від клієнта, розбираються за допомогою кінцевих автоматів. Розібрані запити обробляються поступу в черзі модулів, що задаються конфігураційно. Клієнтські відповіді зберігаються у

буферах, які є сховищем даних в пам'яті чи вказують на адресу в файлах. Буфери об'єднуються в ланки, які визначають послідовність, в них буде передано дані для клієнта.

У випадку, якщо ОС підтримує ефективні операції вводу-виводу, такі як `writew` та `sendfile`, `nginx` застосує їх при наявності та нагоді. Конфігурації `Nginx` розділяються на віртуальні сервери, які в свою чергу діляться на локації. Віртуальний сервер підтримує задання адреси, порта, на яких буде відбуватись прийом з'єднань, а також імена, що містять в собі позначення послідовностей у ланці, в першій та останній частині, також є можливість задати все регулярним виразом. Місцезнаходження, так звані локації, можуть бути задані точно, частково або регулярним виразом. Локації можуть налаштовуватись для обслуговування зі статичного файлу, в `hosts` файлі чи `Memcached`-сервері. Для найбільш ефективного використання пам'яті, `Nginx` використовує пули. Пулами називають послідовність блоків динамічної пам'яті, виділених попередньо. Блок має довжину від 1 до 16 кілобайт. В самому початку, є тільки один пул на один блок. Блоки, в свою чергу, розділяє себе на зайняту та не зайняту область. Дрібні об'єкти виділяються шляхом пересування вказівника на вільну область із врахуванням балансування. Якщо вільного місця не вистачає у всіх блоках одночасно, то буде виділено новий. В разі, якщо блока не достатньо та перевищено значення констант, що обмежують довжину блоку, то місце буде виділено із купи.

Завдяки такому підходу, об'єкти виділяються дуже швидко і мають не великі витрати, крім випадку із вирівнюванням. Сервер також містить у собі функціонал для визначення географічного розпізнавання IP-адрес. В його склад входять база даних відповідностей географічних регіонів, представлена у вигляді дерева (стиснений бор або префіксне дерево) в оперативній пам'яті. Розробники `Nginx` вдало передбачили попереднє розподілення перших кількох рівнів дерева, вони займають рівно одну сторінку пам'яті.

4.4 Особливості архітектури REST API

Архітектурний стиль REST – це підхід до розробки програмного забезпечення, який являє собою набір правил, порад та обмежень, що слід використовувати при розробці веб-служб. Сервіси, що мають повну відповідність цим правилам, іменуються як RESTful веб-сервіси, вони забезпечують сумісність між комп'ютерними системами в мережі Інтернет. RESTful дозволяє системам-клієнтам отримувати доступ до зручного текстового представлення веб-ресурсів і можливість керувати ними, використовуючи попередньо зазначений єдиний набір команд для роботи із інформацією без попередньої автентифікації. Аналогічні підходи, наприклад, SOAP, можуть виставляти власні динамічні набори інформації.

В RESTful-служби, зроблені на певну URI адресу, відповідають на виклик корисною інформацією, у форматі XML, HTML або JSON. У відповіді можна дізнатись що зміни було внесено до системи, або відповідь може забезпечити текстові посилання на сховища ресурсів або пов'язаних із ними даних. При використанні HTTP доступні операції GET, POST, DELETE, PUT та інші попередньо визначені методи CRUD.

Використання такого роду протоколів становго характеру та предвизначені операції дозволяють RESTful системі бути продуктивною, надійною та здатною швидко зростати. Також головною перевагою є можливість масштабування та перевикористання компонентів, з можливістю швидкого оновлення без впливу на вже існуючі системи в цілому, навіть під час процесу, коли програма є запущеною.

Термін репрезентативного стану приведено та визначено Рой Філдінгом в докторській дисертації двохтисячного роцу. Він призначався для виявлення того, як себе вестимуть добре продумані веб-програми, які будуть представляти собою веб-ресурс (віртуальний машинний стан), де користувач буде проходити через додаток використовуючи посилання, такі як account і так операції як GET чи UPDATE, внаслідок чого наступний ресурс передається користувачу у відповідному оновленому стані для подальшого користування.

Архітектурний стиль REST впливає на обмеження наступних архітектурних властивостей:

- ефективність взаємодії компонентів, що є важливим фактором в продуктивності мереж;
- простота єдиних інтерфейсів;
- очевидність зв'язків між компонентами;
- стійкість до збоїв системи в цілому, якщо вийшов із ладу один із компонентів;
- модифікованість компонентів, навіть у разі роботи програми;
- масштабованість, яка дозволяє організувати продуктивну та ефективну взаємодію компонентів.

Філдінг описував вплив REST на можливість розширення, як відокремлення зав'язки клієнт-сервер завдяки спрощенню реалізації компонентів та підвищення масштабованості чистих серверних компонентів. Багатошаровість систем дозволяє вплинути на ефективність та продуктивність спільного кешування. REST надає можливість здійснення проміжних обробок та направлення, у той час як стандартні методи допомагають йому в цьому для позначення семантики та обміну інформацією.

RESTful система визначається шістьма основними правилами. Ці правила обмежують способи, за допомогою яких сервер може обробляти та реагувати на запити користувачів. Виходячи із таких вимог, отримуємо бажані функціональні переваги у вигляді продуктивності, масштабованості, простоти та легкості в модифікації, портативності та надійності. Порушивши хоча б один із принципів – система не може вважатись RESTful.

Специфікація REST складається із наступних правил:

- Розділення відповідальності. Дозволяє розділеним частинам чи компонентам розвиватись самостійно, тим самим гарантуючи організаційні області в мережі;

— Клієнтський контекст знаходиться на сервері між запитами. Стан сеансу залишається на стороні клієнту. Стан сеансу також може бути переданий до інших сервісів, таких як база даних, для підтримування стійкого стану протягом визначеного періоду часу;

— Клієнти та посередники мають можливість кешувати відповіді на запити. Таким чином, буде гарантовано що клієнт не отримає неправильні застарілі дані у відповідь на подальші дії у випадку використання старого посилання;

— Клієнт не може визначити, по якій ієрархії він зв'язаний із сервером – напряду чи через посередників. Посередники можуть покращувати масштабованість, наприклад, при використанні балансувальника. Також це дозволяє покращити безпеку вцілому;

— Сервери мають змогу розширювати або налаштовувати функціонал клієнта, передаючи йому виконуваний код. Наприклад, JavaScript сценарії зі сторони браузера;

— Обмеження інтерфейсу. Важливий фактор при розробці будь-якої RESTful системи, що допомагає спростити чи відокремити архітектуру. Уніфікація інтерфейсу полягає у тому, що індивідуальні ресурси будуть ідентифікуватись в запитах з використанням URI у веб-службах. Самі ресурси, в свою чергу, концептуально відокремлені від їх представлень, що повертаються до клієнту. Наприклад відповіді, що отримує клієнт в HTML, XML або JSON не матимуть жодних схожостей із внутрішнім представленням серверу. Також важливим моментом є те, що клієнт має володіти всіма даними, що потребуються для управління ресурсами. Кожна отримана відповідь має давати змогу у повній мірі обробити інформацію на рівні клієнта.

Отримавши доступ до початкового URI RESTful застосунку, клієнт матиме змогу динамічно користуватись послугами, та викликати саме той функціонал системи, який йому потрібен. На стороні клієнта не варто використовувати жорстке кодування з інформацією про сервіс.

На відміну від протоколу SOAP, в якому є офіційний стандарт по реалізації, REST являє собою архітектурний стиль і чітких обмежень немає. REST не є стандартом, але в ньому використовуються різноманітні стандарти, такі як HTTP, JSON, XML тощо.

4.5 Опис моделі даних системи

Збереження даних в даних у системі є важливим питанням, саме тому було вирішено використовувати базу даних MS SQL, що є найкращим компаньоном для зв'язки .NET Core та Angular. Це дозволить зберігати дані про стан системи, після аналізу та генерації результатів із рейтингами викладачів кафедри.

Велику роль в прийнятті такого рішення зіграв той факт, що рішення було написано на мовах, розроблених компанією Microsoft. Дані в системі будуть динамічно оновлюватись та доповнюватись. Платформа .NET Core та ORM Entity Framework Core найкраще працює із MS SQL так як потреба в сторонніх адаптерах для баз даних відпадає автоматично.

Дані між клієнтом і сервером будуть передаватись у форматі JSON. Цей формат є текстовим файлом для обміну даними комп'ютерами, та може легко бути прочитаним людиною. Формат дозволяє легко описувати сутності, об'єкти та багато різних структур даних. Головне призначення якого, передача структурованої інформації по мережі Інтернет. Формат було розроблено і застосовано Дугласом Крокфордом.

JSON знайшов свою популярність у написанні веб-застосунків, а саме при використанні у парі із такою технологією як AJAX, що виступає як заміна XML під час передачі структурованої інформації між серверами та клієнтами. Основною перевагою над XML є те, що в ньому можна вставляти складні структури до атрибутів, це допомагає зменшити об'єм даних, які передаються, та швидка їх інтерпретація за допомогою JavaScript в об'єкти.

Такий тип даних з'явився у результаті пошуку зручного формату даних для обміну із сервером у реальному часі для додатків що у 2000-х роках були розроблені на основі flash-технологій.

Дуглас Крокфорд саме той, хто активно намагався розробити максимально зручний формат. Він, разом із своїми колегами, хотів створити такий формат, який би використовував можливості браузера для його читання, та забезпечував постійний двусторонній зв'язок клієнта із веб-сервером. Вперше JSON використали у проекті для Cartoon Network, проект дозволяв обмінюватись інформацією із сервером і одночасно маніпулювати елементами інтерфейсу.

Веб-сайт JSON.org був запущений в 2002 році. Через чотири роки, Yahoo! вже почав переводити деякі із своїх веб-сервісів на роботу із цим форматом. Google взяли до роботи із форматом GData у 2006 році.

Структури, на яких базується JSON:

- Пари назва/значення. У порівнянні із мовою C# це можна асоціювати із хеш таблицею, або словником (асоціативним масивом);
- Упорядкований список зі значеннями. Мова C# надає нам аналоги у вигляді масиву або списку;

JSON є універсальною структурою даних. Фактично всі сучасні високорівневі мови підтримують її із самого початку. І не дивно, що саме він використовується для обміну даними між різними мовами програмування, адже є сенс спілкуватися через універсальні структури.

Синтаксичні особливості JSON виділяються у такий формі:

- Об'єктом називають послідовність назв/значень. Об'єкт об'являється символом { та закінчується символом }. Значення слідує за двокрапкою, а пари відділяються комами;
- Масив є послідовністю значень. Починається символом [і закривається символом]. Значення також відділяються комами;

— Саме ж значення може бути строкою в лапках, числом, логічним true чи false, null, об'єктом чи масивом. Всі структури можуть бути вкладеними один в один;

— Строка є послідовністю із нуля або більше символів юнікод, обмежується лапками, одинарними або подвійними, із використанням послідовностей що пишуться зі зворотної косої. Символи представлені у вигляді простого рядка.

Тип строка (String) і число є дуже схожими на аналоги у мові C#.

При відображенні даних, структура є основною для передачі даних у процесі роботи інформаційної системи. Завдяки ній всі сервіси та компоненти системи будуть отримувати інформації та виконувати її обробку.

4.6 Опис компонентів клієнтської та серверної складової

Розроблюваний веб-застосунок складається із двох компонентів, що пов'язані між собою за допомогою прошарку REST API. Перший компонент є клієнтським веб-додатком, який відповідає за презентацію програми для користувача, відображення даних та контролю функціоналу, що доступний користувачеві. Другою частиною є сервер, що відповідальний за обробку внесеної інформації та надання інформації клієнтському додатку.

Оскільки основною перевагою розроблюваної системи є розширюваність та високонавантажуваність, то серверна частина має архітектуру зображену на рисунку 4.1.

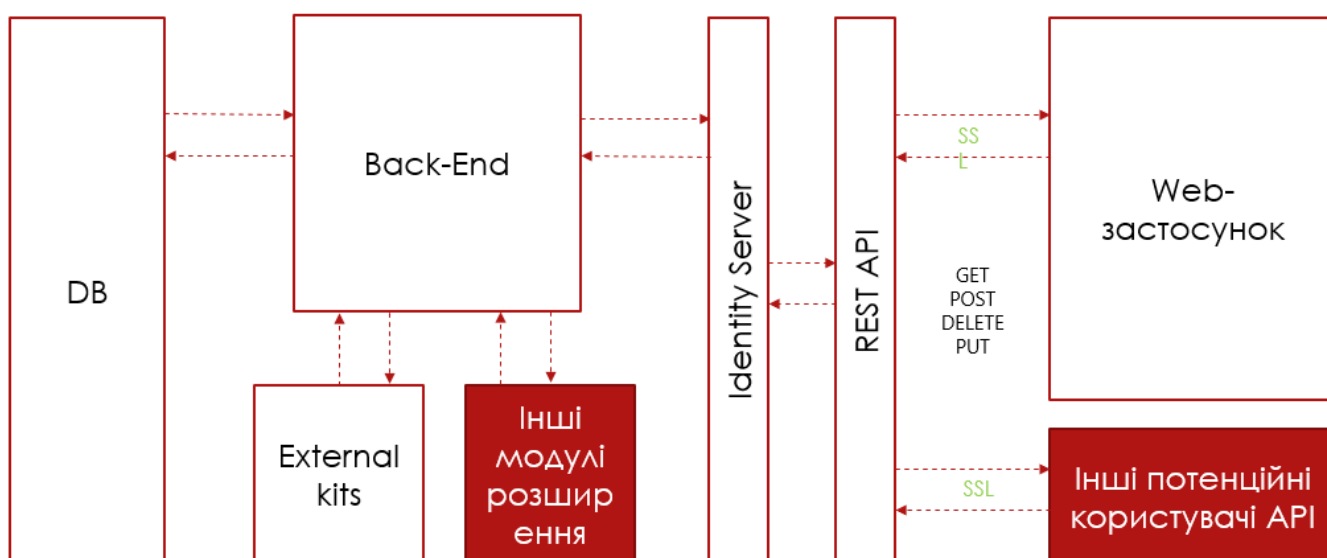


Рисунок 4.1 — Компоненти системи та їх зв'язки

Web-застосунок та інші потенційні клієнти API можна віднести до агрегуючих сервісів та сервісів користувацького інтерфейсу.

База даних, бекенд, Identity Server, в свою чергу, відноситься до серверної частини.

Користувацький інтерфейс представляє собою web-застосунок, який можна відкрити за допомогою будь-якого сучасного браузера. Технологічний набір для розробки даного програмного забезпечення несе у собі такі технології як Angular 8, TypeScript, SignalR та SASS. Перша була створена для написання графічного інтерфейсу і можливості відображення програми на будь-якій платформі у ідентичному вигляді без зовнішніх змін у графіці чи функціоналі. TypeScript було використано для написання модулів, що працюють із серверною частиною та організації правильної взаємодії інтерфейсу з ним. SignalR слугує проміжним сервісом, що дозволяє передавати на клінтську сторону пуш-повідомлення.

В загальному висновку такий набір технологій дозволить користувачеві взаємодіяти із системою, вчасно та справно отримувати оновлення інформації про її стан.

Компонент, що являється точкою збору інформації про вже зібрану та згенеровану рейтингову інформацію системи є агрегуючим. Саме цей сервіс надає змогу відправляти структуровану інформацію, яка отримується із реплікацій сервісів на компонент інтерфейсу користувача, де і відбувається генерування візуальної репрезентації даних з типу JSON в графічне відображення у вигляді графіків та таблиць. Агрегуючий сервіс займається збором інформації по запитах по фрагментам (під фрагментом розуміється конкретний набір статистик за певний період часу чи кількість), після чого він агрегує дані у один великий об'єкт для збереження даних. На запит клієнта, сервер відправляє найновіші дані, для показу їх користувачеві.

Мікросервіс, що обраховує рейтинг викладачів, є компонентом моніторингу. Саме цей компонент відповідає на стороні backend за опрацювання введених даних в реальному часі. Після отримання нових даних для обробки, даний сервіс додає його безпосередньо до бази даних, оновлені дані відправляються агрегуючим сервісом для забезпечення користувача актуальними даними.

4.7 Графічний інтерфейс користувача

У даному підрозділі описаний графічний інтерфейс, який надається користувачу графічним компонентом розглянутим раніше у розділі опису компонентів системи.

При запуску програми користувача зустрічає вікно входу в програму, як на рисунку 5.1.

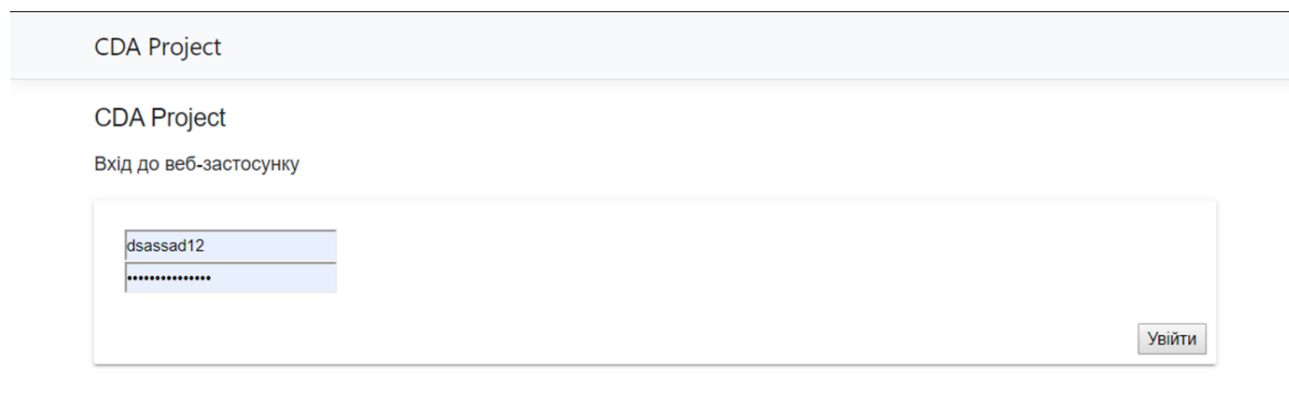


Рисунок 4.2 — Сторінка входу до системи

Після входу до системи користувач потрапляє на домашню сторінку, зображену на рисунку 4.3, на якій знаходиться інструкція користувача та стартові елементи програми.

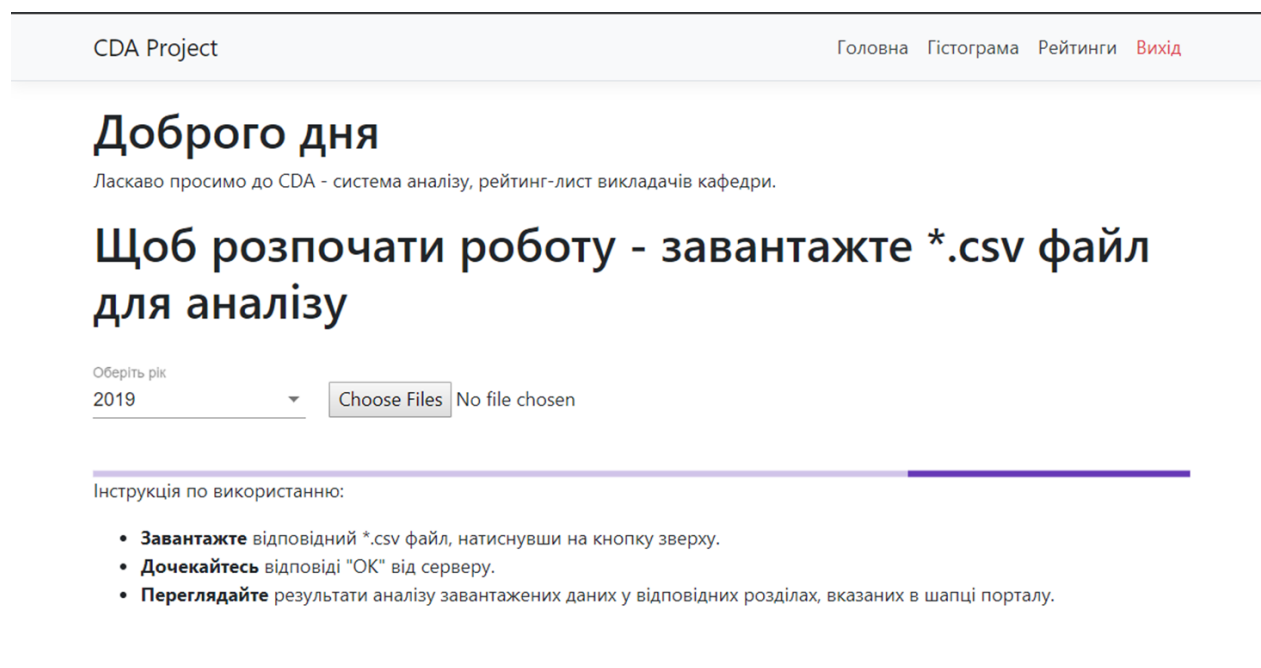


Рисунок 4.3 — Демонстрація домашньої сторінки

У верхній частині вікна знаходиться меню навігації по додатку та кнопка виходу із системи. Після виконання аналізу введених даних з'являється повідомлення про завершення роботи. Користувач може приступити до користування веб-застосунком.

Оскільки програма зберігає обраховані дані у базі даних, можна переглядати гістограми за різні роки, показано на рисунку 4.4.

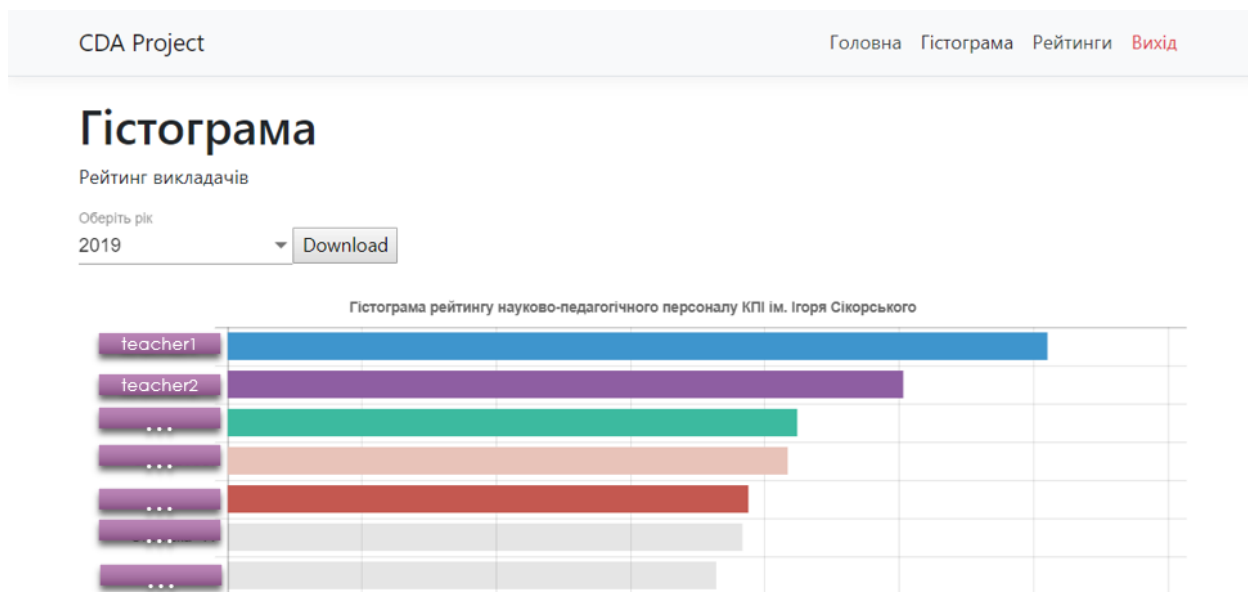


Рисунок 4.4 — Вікно гістограм

Наступна сторінка, рейтинг, дозволяє переглянути різні рейтинги викладачів, що є вибірками за різними параметрами, показано на рисунку 4.5.

CDA Project Головна Гістограма Рейтинги Вихід

Рейтинги викладачів

Оберіть бажану порівняльну статистику

Оберіть рік
2019

Топ-10 в загальному рейтинзі Топ по Науково-інноваційній роботі Топ по Організаційно-виховній роботі

Фільтрувати

ПІБ	R(н-м)	R(н-і)	R(о-в)	Сумма
teacher	977.086	4568	560	6105.086
teacher	1682.6	1010	2340	5032.6
teacher	2380.25	1068	800	4248.25
teacher	1571.88	500	2100	4171.88
teacher	1864.031	1782.525	230	3876.556
teacher	2801.688	670	365	3836.688

Рисунок 4.5 — Сторінка із рейтингами

Переглянути результати можна експортувати у форматі PDF та завантажити, як показано на рисунку 4.6.

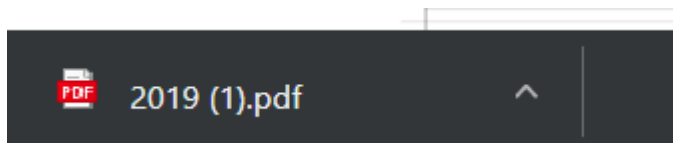


Рисунок 4.6 — Завантажений згенерований

При відкритті файлу користувача чекає гістограма, та всі таблицьки, що були згенеровані на сторінці рейтингів, як показано на рисунок 4.7 і рисунку 4.8.

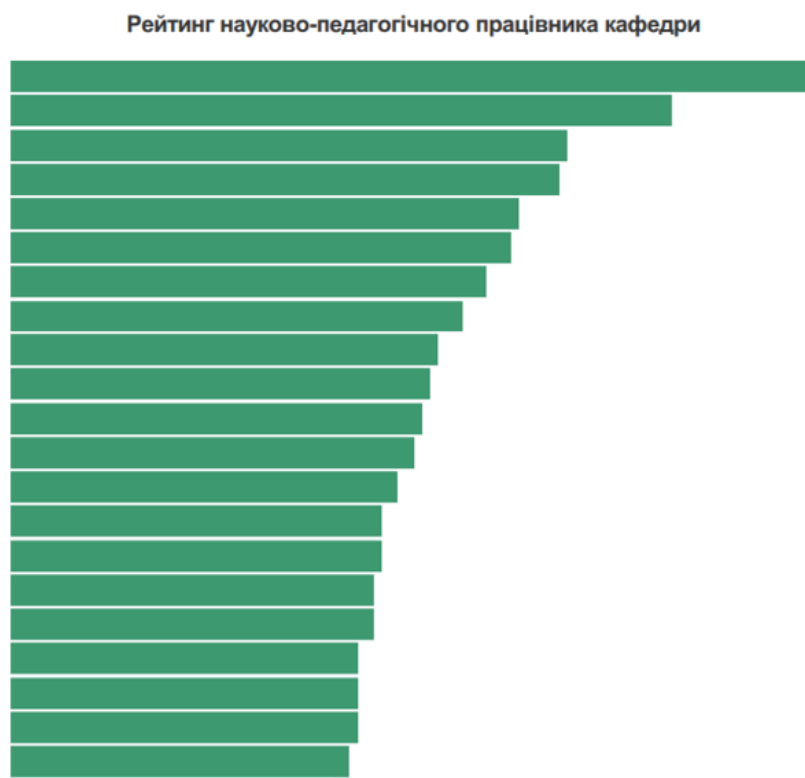


Рисунок 4.7 — Гістограма в згенерованому звіті

Кращі викладачі по загальному рейтингу

ПІБ	R(н-м)	R(н-і)	R(о-в)	Рейтинг, R
	977.086	4568	560	6105.086
	1682.6	1010	2340	5032.6
	2380.25	1068	800	4248.25
	1571.88	500	2100	4171.88
	2801.688	670	365	3836.688
	2211.428	1147.5	280	3638.928
	2458.65	450	516	3424.65
	1978.94	764	535	3277.94
	1670.5	645	810	3125.5
	2001.9	502	575	3078.9

Рисунок 4.8 — Приклад згенерованої таблиці у експортованому файлі

5. РОЗРОБКА СТАРТАП ПРОЕКТУ

Стартап – це нещодавно створена компанія, яка знаходиться лише на стадії розвитку, але має інноваційні ідеї. Саме ці нововведення допоможуть перевершити конкурентів та вийти на ринок з новим продуктом, послугою. Основна особливість стартапів – недостатність фінансів, пошук «бізнес-ангелів» та відношення не тільки до IT-сфери, а і до будь-якої області ринку. Зараз стартап-проекти можна називати венчурними.

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту. В межах цього етапу:

- розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються ринкові можливості щодо його реалізації;
- на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

5.1 Опис ідеї проекту

Далі було проаналізовано і подано у вигляді таблиць:

- зміст ідеї, що пропонується;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- відмінність від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці 5.1 і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка веб-застосунку для системи аналізу рейтингу викладацького складу кафедри	Впровадження застосунку у якості робочого інструменту кафедри АПЕПС	Аналіз рейтингу викладацького складу кафедри завжди доступний всюди і у будь-який час, де є пристрій з браузером та доступом до мережі інтернет.
	Масштабування застосунку в рамках всього факультету або університету	У зв'язку зі збільшеною вибіркою з'являється можливість проводити нові види аналізу відповідно до вимог факультету.

Продовження таблиці 5.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Гнучка архітектура	Інтеграція в застосунок інших інформаційних ресурсів університету (внутрішніми та зовнішніми).	Користувач може отримати всю потрібну йому інформацію швидко і в одному місці.
Модульність системи	Розширення функціоналу програми відповідно до вимог користувача.	Відповідно з вимогами користувача, функціонал може бути розширений за короткий період часу.
Відкритий програмний інтерфейс	API	Користувач має документацію до API застосунку, і у разі наявності відповідних прав для користування – може інтегрувати його до своїх рішень.

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 5.2).

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

Таблиця 5.2. Визначення сильних, слабких та нейтральних характеристик проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Веб-застосунок для аналізу рейтингу викладацького складу кафедри.	Програмний продукт що використовується зараз (система обліку).			
1.	Доступ до програми	Через браузер, у будь-який час.	Продукт доступний тільки один тиждень, два рази на рік.			+
2.	Простота інтерфейсу	Користувач не перевантажується, так як весь функціонал поділений по розділам що з'єднуються зручною навігацією.	Інтерфейс перевантажений непотрібною для користувача інформацією.		+	

Продовження таблиці 5.2

№ п/п	Техніко- економічні характеристи ки ідеї	Веб-застосунок для аналізу рейтингу викладацького складу кафедри.	Програмний продукт що використовується зараз (система обліку).	W (сла бка сто рон а)	N (нейтр альна сторон а)	S (сильн а сторон а)
3.	Розширення функціоналу	Можливе.	Продукт в стадії підтримки, розробка завершена.			+
4.	Інтеграція зі сторонніми ресурсами	Присутня.	Відсутня.		+	
5.	Можливість масштабуван ня	Передбачено в архітектурі.	Вже масштабовано.		+	
6.	Не вибагливість до ресурсів комп'ютера користувача.	Працює всюди, де є браузер.	Система Windows XP, або підтримка запуску програм в режимі сумісності із цією операційною системою.			+
7.	Розміщення програмного продукту	Cloud: Azure, AWS або виділений сервер.	Комп'ютер користувача.		+	

5.2 Технологічний аудит ідеї проекту

Було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 5.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту.

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як React.js на платформі Node.js через їх доступність та безкоштовність. Також було використано сервер балансування NGINX для забезпечення висконавантажуваності системи.

Таблиця 5.3. Технологічна здійсненність проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Інтерфейс користувача.	Розробити веб-сайт та UX-дизайн системи.	Angular 8, TypeScript.	Так.
2.	Серверна частина із власним API.	Розробити серверну частину із власною базою даних та програмного інтерфейсу для доступу до них.	.Net Core WebAPI.	Так.
3.	Розробка аналітичного функціоналу який пропонують аналогічні системи.	Проаналізувати функціонал попередніх систем. Зібрати вимоги до системи у користувачів попередньої системи.	Наявне.	Так.

Продовження таблиці 5.3

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
4.	Зворотна підтримка вводу даних.	Створити документ, шаблон, який повторює структуру файлів, що завантажувались до попередніх систем.	Наявне.	Так.
5.	Інтеграція із сторонніми системами.	Найняти розробників, написати ТЗ, надати їм документацію даного проекту.	Наявне.	Так.
6.	Розміщення продукту в мережі інтернет.	Виділити локальний сервер або зняти хостинг.	Наявне: Linux-сервер із підтримкою Docker та одним гігабайтом оперативної пам'яті.	Так.
7.	Розробка документації для сторонніх та внутрішніх розробників.	Використання swagger, детальний опис змін у програмному коді в GitHub.	Наявне: всі зміни детально записуються з коментарями до історії змін проекту, та легкі до читання.	Так.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1.	Кількість головних гравців	Більше двох
2.	Загальний обсяг продаж	Більше \$1 млн
3.	Динаміка ринку	Зростає
4.	Наявність обмежень для входу	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Відповідно до вимог Міністерства Освіти і Науки
6.	Середня норма рентабельності в галузі (або по ринку)	8%

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 5.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та зформовано орієнтовний перелік вимог до товару для кожної групи (таблиця 5.5).

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Економія часу	Викладачі, працівники кафедри	Потребують доступ в систему в будь-який час та невибагливість програмного продукту до ресурсів комп'ютера, з якого ведеться робота.	До продукції: якість До компанії: надійність
2.	Отримання продукту	Кафедра, університет	Потребують широких можливостей регулювання умов збуду продукції	Економія коштів, отримання прибутку
3.	Збільшення потоку користувачів	Кафедра, університет	Потребують функціональності, масштабованості, можливості інтегруватись до інших систем.	Збільшення числа користувачів та потенційних клієнтів

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6, 5.7).

Таблиця 5.6. Фактори загроз

№ п/п	Фактори	Зміст загрози	Можлива реакція компанії
1.	Запуск і супровід	Можливі непередбачувані проблеми.	Найняти команду підтримки.
2.	Надійність	Сервіс повинен забезпечувати безвідмовну роботу, проблема пікового навантаження.	Орендувати потужний сервер.
3.	Розробка ПЗ	Можливі труднощі при розробці, відставання від термінів.	Вирішити проблеми, найняти додаткових працівників, змістити терміни.
4.	Розташування серверу	Оптимізація роботи веб-застосунку. Локація датацентру. Сервіс максимально ефективний на у співвідношенні потрібних ресурсів до одного користувача.	Обрати сервер що географічно розташований ближче до основної маси користувачів.
5.	Захист інформації	Можливі проблеми при інтеграції до сторонніх сервісів або ресурсів.	Розробити стратегію аудиту ризиків при інтеграції.

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Ресурси	Економія ресурсів, часу	Забезпечити користувачів необхідним функціоналом
2.	Прибуток	Отримання прибутку	Заходи для збільшення рентабельності

Надалі було проведено аналіз пропозиції: визначено загальні риси конкуренції на ринку (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції: Монополія	Аналогічне рішення вже існує, і запроваджене в роботу.	Забезпечити користувачів ширшими функціональними можливостями для аналізу та перегляду даних.
За рівнем конкурентної боротьби: Локальний	Рішення є локальним, на рівні університету.	Забезпечити максимальну доступність та швидкість, де це можливо.
За галузевою ознакою: внутрішньогалузева	Можлива конкуренція із вже існуючими програмними рішеннями.	Жорстка конкуренція. Необхідно зробити послуги кращими ніж в конкурентів.

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Конкуренція за видами товарів: товарно-родова	Можлива конкуренція між вже впровадженими аналогічними сервісами.	Необхідно зробити послуги кращими та зручнішими ніж в конкурентів.
Характером конкурентних переваг: ціновий	Цінова – зробити систему аналізу більш гнучкою та економною для кінцевого власника.	Цінова – автоматизувати процеси, які робляться вручну, тим самими зробити їх дешевшими.
Інтенсивність: марочна	Конкуренція за надання послуг університету.	Розробка та набір популярності власного ПЗ.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (таблиця 5.9).

За результатами аналізу таблиці 5.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок було враховано при формулюванні переліку факторів конкурентоспроможності далі. На основі аналізу конкуренції, наведеного в таблиці 5.9, а також з урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6, 5.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у таблиці 5.10.

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямий конкурент в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замітники
	Система яку наразі використовує НТУУ «КПІ ім. Ігоря Сікорського»	Бар'єри входження на ринок: зв'язки з правовласниками, отримання дозволу на обробку персональних даних, розмір капіталовкладень	Фактори сили постачальників: змінні витрати на датацентри, технології на яких базується веб-застосунок є безкоштовними	Фактори сили: змінні витрати, високий рівень чутливості до зміни цін на електроенергію, контроль якості	Фактори загроз: ціна обслуговування, лояльність замовника
Висновки	Середня інтенсивність конкурентної боротьби.	Є можливості входу на ринок. Є потенційні конкуренти.	Правовласники інструментів розробки не накладають обмеження на функціонал ПЗ, але це може виникнути в перспективі.	Для клієнтів найбільш важлива цінова доступність послуг, захист персональних даних.	Обмеження економічної рентабельності розвитку веб-застосунку.

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Науково-технічний потенціал.	Веб-застосунок дозволяє проводити аналіз наукової роботи працівників в короткі строки.
2.	Фінансово-економічний потенціал.	Скорочення часу на проведення аналізу та формування висновків, економія людино- ресурсів.
3.	Ефективність робочого контролю.	Аналіз ефективності викладацького складу кафедри.
4.	Рівень технічного обслуговування в після виробничий період.	Завдяки правильній архітектурі та детальному тестуванню проекту – обслуговування коштуватиме набагато менше, ніж в конкурентів.

За визначеними факторами конкурентоспроможності (таблиця 5.10) проведено аналіз сильних та слабких сторін стартап-проекту (таблиця 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним продуктом						
			-3	-2	-1	0	+1	+2	+3
1.	Науково-технічний потенціал.	14	+						
2.	Фінансово-економічний потенціал.	16					+		
3.	Ефективність контролю.	18					+		

Продовження таблиці 5.11

№ п/п	Фактор конкурентоспроможності	Бали 1-20	-3	-2	-1	0	+1	+2	+3
4.	Рівень технічного обслуговування в після виробничий період.	20	+						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 5.12. SWOT-аналіз стартап-проекту

Сильні сторони: Технологічність, архітектура, легкість в обслуговуванні після виробничої стадії.	Слабкі сторони: На старті проекту будуть доступні не всі функціональні можливості, не чіткі терміни розробки.
Можливості: Вихід на ринок, економічний потенціал, перспектива у масштабуванні.	Загрози: Конкуренти, обмеження економічної рентабельності, накладення обмежень від правовласників інструментів розробки ПЗ.

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 5.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Пришвидшення процесу розробки, підвищення надійності, кількості, доступної інформації потрібної для користувача.	Малоймовірно на початковому етапі вдасться отримати фінансування, для збільшення штату працівників.	1-3 місяці
2.	Розробка проекту власними силами на початковому етапі.	Немає потреби в значному фінансуванні	5-6 місяців

Для подальшої реалізації було обрано альтернативу №2.

5.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 5.14).

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи користувачів	Готовність користувачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1.	Зав. Кафедри	Середня готовність.	Середній попит.	Низька інтенсивність.	Легко.
2.	Працівники кафедри	Середня готовність.	Низький попит.	Низька інтенсивність.	Легко.
Обрані цільові групи: Зав. Кафедри та працівники кафедри.					

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано товар та визначено стратегію охоплення ринку – стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 5.15).

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурентів, і які?	Стратегія конкурентної поведінки.
1.	Стратегія стабілізації	Стратегія диференційованого маркетингу	Користувачам надається більше можливостей.	Стратегія диференціації

Продовження таблиці 5.15

№ п/п	Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурентів, і які?	Стратегія конкурентної поведінки.
2.	Стратегія зростання	Масовий маркетинг	Нижча вартість послуг за рахунок раціонального використання часу та ресурсів серверу та клієнтського комп'ютера.	Стратегія лідерства по витратах.

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 5.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Аналоги вже є на ринку.	Забирати існуючих споживачів, залучення нових.	Ціна послуг нижча середнього по ринку.	Стратегія виклику лідера.

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 5.5), а також в залежності від обраної базової стратегії розвитку (таблиця 5.15) та стратегії конкурентної поведінки (таблиця 5.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Цінова доступність.	Наступальна стратегія.	Низька ціна за рахунок автоматизації процесів, спрощення доступу до застосунку,	Ціна розробки та підтримки, функціонал, доступність.
2.	Можливість керування послугами та додаванням функціоналу.	Наступальна стратегія.	Архітектура проекту дозволяє швидку модифікацію та додавання функціоналу до веб-застосунку.	

В результаті отримано узгоджену систему рішень щодо ринкової поведінки стартап-компанії.

5.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 5.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1.	Реалізація аналогічного функціоналу у більш зрозумілій для користувача формі.	Користувач може швидко і безболісно перейти на користування новою системою.	<ul style="list-style-type: none"> - Доступність - Функціонал - Інтеграція
2.	Веб-застосунок.	Користувач має змогу користуватись системою із будь-якого пристрою.	
3.	Відкритий API.	Користувачі можуть інтегрувати систему в свої рішення, або розробляти нові види клієнтських додатків.	

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 5.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і/або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язане з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє собою рішення того, як буде реалізовано товар на ринку; включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості, доставка, умови оплати тощо).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
1. Товар за задумом	Веб-застосунок, товар – постачання послуг з аналізу рейтингу викладачів кафедри. Покращення економічних показників. Товар надає можливість робити рутинні задачі швидше, покращує лояльність користувачів та економить часові ресурси.
2. Товар у реальному виконанні	Властивості:
3. Товар із підкріпленням	Веб-застосунок, графічний інтерфейс користувача, API.

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання шляхом патентування. Наступним кроком є визначення цінових меж (таблиця 5.20), якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається

під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів. Аналіз проведено експертним методом. В результаті проведення аналізу встановлено верхню та нижню межі встановлення ціни на товар, що надається користувачам.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на замітники (додаток що зараз використовується на кафедрі)	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	500-5000 гривень на рік	5000 гривень в місяць	всі	300 гривень на місяць.- 1000 гривень на місяць

Наступним кроком є визначення оптимальної системи збуту (таблиця 5.21).

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Прямий збут	Укладення договору із навчальною установою.	Однорівневий канал.	«Розробник – навчальний заклад – працівники кафедри»

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікації, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Навчальний заклад	Інтернет	Персональний продаж	Зацікавити керівництво навчального закладу співпрацювати заради отримання прибутку та покращення сервісу.	Переконати керівництво навчального закладу, що співпраця дозволить налагодити процесі, пришвидшити роботу пов'язану з аналізом рейтингу викладацького складу кафедри.

Результатом аналізу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні

переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновок до розділу 5

Було проведено аналіз програмного продукту з позиції перспективи розвитку стартап-проекту, а саме:

- Проведено аналіз ринку існуючих рішень та потреб потенційних користувачів.
- Вивчено потенційні можливості та ризики розвитку бізнесу за стартап-проекту.
- Запропоновано межі ціно формування на розроблений продукт.
- Визначено стратегії просування стартап-проекту.
- Розроблено маркетингову програму.

ВИСНОВКИ

По завершенню виконання магістерської дисертації було проаналізовано літературу та вже існуюче програмне забезпечення для аналізу та формування рейтингу викладацького складу кафедри, з якого було зроблено здобуто наступну інформацію:

- з чого складається рейтинг кафедри;
- вплив рейтингу викладача на загальний рейтинг;
- оптимізовано інформацію, необхідну для створення автоматизованої системи аналізу викладацького складу.

Було розглянуто існуючі аналогічні системи для аналізу викладацького складу кафедри та сформульовано наступні завдання:

- проаналізувати можливості сучасних систем аналогів;
- аналіз методів опрацювання рейтингів НПП;
- удосконалити використання методів розрахунку порівняно із системами попередниками;
- спроектувати архітектури нової системи;
- розробити спроектовану систему.

Було проведено аналіз властивостей та закономірностей проблем при зборі інформації та формуванні кінцевих рейтингових результатів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алексеев К. Б. Управление космическими летательными аппаратами / К. Б.
2. Алексеев, Г. Г. Бебенин. – М.: Машиностроение, 1974. – 344 с.
3. Алпатов А.П. Динамика космических аппаратов с магнитными системами управления / А.П. Алпатов, В.И. Драновский – М.: Машиностроение, 1978. – 200 с.
4. Бондарь П.М. Физические основы ориентации и навигации. Ч.1. Физические поля Земли / П.М. Бондарь, Ю.В. Степанковский. – К.: Корнейчук, 2002. – 104с.
5. Карпенко С.О. Исследование возможности трехосной стабилизации микро-спутника с использованием только токовых катушек // Современные проблемы фундаментальных и прикладных наук». Часть 7. Управление и прикладная математика / С.О. Карпенко. – Москва.: Долгопрудный, 2006. – С.218 – 219.
6. Коваленко А.П. Магнитные системы управления космическими летательными аппаратами / А.П. Коваленко. – М.: Машиностроение, 1975. – С.248.
7. Козаков А.В. Магнитогироскопическая система – достойный соперник гравитационной системы ориентации на малых спутниках. // Вопросы электромеханики. Труды НПП ВНИИЭМ. Том 100 / А.В. Козаков. – М.: ФГУП «НПП ВНИИЭМ», 2001. С. 156-176.
8. Международное аналитическое поле // Геомагнетизм и аэрономия, 9. –1969. – №3. – С.584 – 586 с.
9. Раушенбах Б.В. Управление ориентацией космических аппаратов. /Б.В. Раушенбах, Е.Н. Токарь. – М.: Наука, 1974. – 600 с.
10. Студопедия. Модульное программирование [Electronic resource]. – Mode of access: <https://studopedia.org/5-63676.html>

11. Блоги Google App Engine [Электронный ресурс]. – Электрон. дан. — 10 вещей, которые вы (возможно) не знали про App Engine. - Режим доступа: <http://habrahabr.ru/blogs/gae/64121/>
12. Min Chen, Shiwen Mao, Yin Zhang, Victor C.M. Leung. Big Data. Related Technologies, Challenges, and Future Prospects. — Springer, 2014. — 100 p
13. Kohavi, Rothleder and Simoudis (2002). "Emerging Trends in Business Analytics". Communications of the ACM. 45 (8): 45–48p
14. David Haertzen. ETL Tools // The Analytical Puzzle: Profitable Data Warehousing, Business Intelligence and Analytics. — Technics Publications, 2012. – 346 с.
15. David Loshin. ETL (Extract, Transform, Load) // Business Intelligence. — 2nd. — Morgan Kaufmann, 2012. – 400 с.
16. Inmon W.H. Building the Data Warehouse, 4th Edition. — Hoboken, NJ: Wiley, 2005. — 576 с.
17. Белоусов А. И., Ткачев С. Б. Дискретная математика. — М.: МГТУ, 2006. — С. 460-587.
18. Флэнаган Д. JavaScript. Карманный справочник. Сделайте веб-страницы интерактивными! / Перевод А.Г. Сысолюк. — Москва.: Издательский дом "Вильямс", 2015. — С. 320.
19. JavaScript. Библия пользователя = JavaScript. Bible / Денни Гудман (Danny Goodman), Майкл Моррисон (Michael Morrison); пер. с англ. И. В. Василенко. — 5-е изд. — Москва, Санкт-Петербург, Киев: Дилектика, 2006. — С. 3, 26. — 1184 с.
20. Green, Brad; Seshadri, Shyam (March 22, 2013). React.JS (1st ed.). O'Reilly Media. p. 150.
21. Стивен Хольцнер. Ajax Библия программиста = Ajax Bible. — М.: Диалектика, 2009. — С. 553.
22. Дейв Крейн, Эрик Паскарелло, Даррен Джеймс. AJAX в действии: технология — Asynchronous JavaScript and XML = Ajax in Action. — М.: Вильямс, 2006. — С. 640.

23. Balalaie, A. (2016-05-01). «Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture». *IEEE Software* 33 (3): 42–52.
24. Rodgers, Peter. "Service-Oriented Development on NetKernel- Patterns, Processes & Products to Reduce System Complexity". *CloudComputingExpo. SYS-CON Media*. Retrieved 19 August 2015.
25. Dilshat Salikhov, Kevin Khanda, Kamill Gusmanov, Manuel Mazzara, Nikolaos Mavridis. "Microservice-based IoT for Smart Buildings". Retrieved 9 Feb 2018.
26. Schonberger, R.J. (2001). *Let's Fix It! Overcoming the Crisis in Manufacturing*. New York: Free Press. pp. 70–71.
27. Петруненко А. Оценка коммерческой привлекательности проекта [Электронный ресурс]. – Режим доступа: <http://www.techbusiness.ru/tb/archiv/number2/page01.htm>
28. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
29. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.
30. Квашнин А. Как управлять портфелем технологий и интеллектуальной собственностью : серия методических материалов «Практические руководства для центров коммерциализации технологий» / под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 60 с.
31. Квашнин А. Как продвигать проекты коммерциализации технологий: серия методических материалов «Практические руководства для центров коммерциализации технологий» / М. Катешова, А. Квашнин, под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 52 с.
32. Филип Котлер, Роланд Бергер, Нильс Бикхофф. Стратегический менеджмент по Котлеру. Лучшие приемы и методы = *The Quintessence of Strategic*

Management: What You Really Need to Know to Survive in Business. — М.: Альпина Паблишер, 2012. — 144 с.

33. Бариленко В. И. Бизнес-анализ как важный вид консалтинговых услуг // РИСК: Ресурсы, Информация, Снабжение, Конкуренция. — № 4. — 2012. — С.202-207.

ДОДАТОК А

Публікації

Аналіз рейтингу викладацького складу кафедри

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТР21224_18М

Аркушів 3

2019